TextConverter

INHERITS FROM	ResultObject
DECLARED IN	TextConverter.h

CLASS DESCRIPTION

The TextConverter class serves as a base class for new classes. It provides little functionality on its own. Classes decended from TextConverter will convert text from one character set to another. For instance, a subclass might convert text from a Macintosh file to equivalent text for a NeXT application. The TextConverter class, itself, merely takes Characters, and returns them to the caller unaltered.

There are two primarly methods provided by TextConverter, and they should be overridden by all subclasses. *ConvertCharacter:* takes a single character, and returns an equivalent character in the destination character set (e.g. if it is passed a Mac 0xA5 (bullet) character, it might return the character 0xB7 (a bullet in the NeXT character set)). If the character can not be mapped to a single character, then the subclass is free to choose whether to leave the value unchanged, or make an approximation. The second method, *ConvertString:WithLength:* also converts from one character set to another. Unlike *ConvertCharacter:*, it allows for

converting a single character to multiple (e.g. a greater than or equal sign in one character set might be mapped to the pair of characters `>=').

This class does not have specialized *init* or *free* methods. The inherited ones work adequately. Note that this class is essentially just a pair of functions bundled up as a fancy class! =)

INSTANCE VARIABLES

Declared in TextConverter none

METHOD TYPES

Converting characters
- ConvertCharacter:
- ConvertString:WithLength:

CLASS METHODS

None

INSTANCE METHODS

ConvertCharacter:

- (Character) **ConvertCharacter:** (Character) *theCharacter*

0 Result codes and test
1 The returned character

This returns a single character in the destination character set that most closely matches the character value that it is provided. In this TextConverter class, this method merely returns the value it was passed.

ConvertString:WithLength:

- (Pointer) **ConvertString:** (Pointer) *theData* **WithLength:** (Integer) *length*
    0 Result codes and test
    1 The returned Pointer
    2 The lenght of the returning data

This method takes a pointer to a string of *length* characters (*length* should be a character count, not a byte count, should the two ever be different). Like **ConvertCharacter:**, this will convert characters in the source data, which are assumed to be in one character set, into characters in its destination character set. The TextConverter class merely maps each incomming character to an identical outgoing character. There are two things to note in particular about this method. This method allows for mapping multiple source characters to single destination characters, or single source to multiple destination characters. Because of this, it can not directly modify the source string. Consequently, it always allocates a new block of memory, and converts into it. A pointer to this new block is returned, along with the length of the converted data. This data can later be deallocated with FreePointer().

BUGS AND PROBLEMS

none yet

ENHANCEMENT IDEAS

none

CONSTANT, DEFINED TYPES AND ERROR CODES

```
#define errCANTMAPTOONE         1001   /* set by ConvertChar: if mapping would be > 1 dest char */
#define errCANTMAPFROMONE       1002   /* set by ConvertChar: if mapping needs > 1 source char */
```

MODIFICATION HISTORY

$Log: TextConverter.rtf,v $Revision 1.2 93/04/04 23:45:24 deathSun Apr 4 23:45:23 PDT 1993Revision 1.1 93/01/10 15:08:53 deathSun Jan 10 15:08:53 PST 1993